

```

LALFrDetector
CHAR      name[LALNameLength];
REAL8     vertexLongitudeRadians;
REAL8     vertexLatitudeRadians;
REAL4     vertexElevation;
REAL4     xArmAltitudeRadians;
REAL4     xArmAzimuthRadians;
REAL4     yArmAltitudeRadians;
REAL4     yArmAzimuthRadians;

```

```

OverlapReductionFunctionParameters
UINT4     length; /* length of vector containing overlap function */
REAL8     f0; /* start frequency */
REAL8     deltaF;

```

```

void LALOverlapReductionFunction( LALStatus *status,
REAL4FrequencySeries *output,
const LALDetectorPair *detectors,
const OverlapReductionFunctionParameters *parameters);

```

```

void LALCreateDetector( LALStatus *status,
LALDetector *output,
const LALFrDetector *input,
const LALDetectorType type );

```

```

LALDetector
REAL8     location[3];
REAL4     response[3][3];
LALDetectorType type;
LALFrDetector frDetector;

```

```

LALDetectorPair
LALDetector detectorOne;
LALDetector detectorTwo;

```

```

LALNumCachedDetectors[]
LALDetectorIndexLHODIFF,
LALDetectorIndexLLODIFF,
LALDetectorIndexVIRGODIFF,
LALDetectorIndexGEO600DIFF,
LALDetectorIndexTAMA300DIFF,
LALDetectorIndexCIT40DIFF,

```

```

StochasticOmegaGWParameters
REAL4     alpha;
UINT4     length;
REAL8     f0; /* start frequency */
REAL8     deltaF; /* frequency spacing */
REAL8     fRef; /* reference frequency */
REAL4     omegaRef;

```

```

void LALStochasticOmegaGW( LALStatus status,
REAL4FrequencySeries *output,
const StochasticOmegaGWParameters *parameters);

```

```

StochasticInverseNoiseInput
REAL4FrequencySeries *unCalibratedNoisePSD ;
COMPLEX8FrequencySeries *responseFunction;

```

```

void LALStochasticInverseNoise( LALStatus *status,
StochasticInverseNoiseOutput *output,
const StochasticInverseNoiseInput *input);

```

```

StochasticInverseNoiseOutput
REAL4FrequencySeries *calibratedInverseNoisePSD;
COMPLEX8FrequencySeries *halfCalibratedInverseNoisePSD;

```

```

StochasticOptimalFilterNormalizationInput
REAL4FrequencySeries *overlapReductionFunction;
REAL4FrequencySeries *omegaGW;
REAL4FrequencySeries *inverseNoisePSD1;
REAL4FrequencySeries *inverseNoisePSD2;

```

```

void LALStochasticOptimalFilterNormalization( LALStatus *status,
StochasticOptimalFilterNormalizationOutput *output,
const StochasticOptimalFilterNormalizationInput *input,
const StochasticOptimalFilterNormalizationParameters *parameters);

```

```

REAL4WithUnits
REAL4 value;
LALUnit units;

```

```

StochasticOptimalFilterNormalizationParameters
REAL8     fRef;
BOOLEAN   heterodyned;
REAL4Vector *window1;
REAL4Vector *window2;

```

```

StochasticOptimalFilterNormalizationOutput
REAL4WithUnits *normalization;
REAL4WithUnits *variance;

```

```

StochasticOptimalFilterInput
REAL4FrequencySeries *overlapReductionFunction;
REAL4FrequencySeries *omegaGW;
COMPLEX8FrequencySeries *halfCalibratedInverseNoisePSD1;
COMPLEX8FrequencySeries *halfCalibratedInverseNoisePSD2;

```

```

void LALStochasticOptimalFilter( LALStatus *status,
COMPLEX8FrequencySeries *optimalFilter,
const StochasticOptimalFilterInput *input,
const REAL4WithUnits *lambda);

```

